



469.00

201

08/533706

Virus Detection and Removal Apparatus for Computer NetworksBackground of the Invention1. Field of the Invention

The present invention relates generally to computer systems and computer networks. In particular, the present invention relates to a system and method for detecting and removing computer viruses. Still more particularly, the present invention relates to a system and method for detecting and removing computer viruses from file and message transfers between computer networks.

2. Description of the Related Art

During the recent past, the use of computers has become widespread. Moreover, the interconnection of computers into networks has also become prevalent. Referring now to Figure 1, a block diagram of a portion of a prior art information system 20 is shown. The portion of the information system 20 shown comprises a first network 22, a second network 24 and third network 26. This information system 20 is provided only by way of example, and those skilled in the art will realize that the information system 20 may include any number of networks, each of the networks being its own protected domain and having any number of nodes. As shown in Figure 1, each of the networks 22, 24, 26 is formed from a plurality of nodes 30, 32. Each of the nodes 30, 32 is preferably a microcomputer. The nodes 30, 32 are coupled together to form a network by a plurality of network connections 36. For example, the nodes 30, 32

1 may be connected together using a token ring format, ethernet format or any of
2 the various other formats known in the art. Each of the networks 22, 24, 26
3 includes a node 32 that acts as a gateway to link the respective network 22, 24, 26
4 to other networks 22, 24, 26. Each of the gateway nodes 32 is preferably coupled by
5 a standard telephone line connection 34 such as POTS (Plain Old Telephone
6 Service) or a T-1 link to the other gateway nodes 32 through a telephone
7 switching network 28. All communication between the networks 22, 24, 26 is
8 preferably performed through one of the gateway nodes 32.

9 One particular problem that has plagued computers, in particular
10 microcomputers, have been computer viruses and worms. A computer virus is
11 a section of code that is buried or hidden in another program. Once the program
12 is executed, the code is activated and attaches itself to other programs in the
13 system. Infected programs in turn copy the code to other programs. The effect of
14 such viruses can be simple pranks that cause a message to be displayed on the
15 screen or more serious effects such as the destruction of programs and data.
16 Another problem in the prior art is worms. Worms are destructive programs
17 that replicate themselves throughout disk and memory using up all available
18 computer resources eventually causing the computer system to crash.
19 Obviously, because of the destructive nature of worms and viruses, there is a
20 need for eliminating them from computers and networks.

21 The prior art has attempted to reduce the effects of viruses and prevent
22 their proliferation by using various virus detection programs. One such virus
23 detection method, commonly referred to as behavior interception, monitors the

1 computer or system for important operating system functions such as write,
2 erase, format disk, etc. When such operations occur, the program prompts the
3 user for input as to whether such an operation is expected. If such an operation
4 is not expected (e.g., the user was not operating any program that employed such
5 a function), the user can abort the operation knowing it was being prompted by a
6 virus program. Another virus detection method, known as signature scanning,
7 scans program code that is being copied onto the system. The system searches for
8 known patterns of program code used for viruses. Currently, signature scanning
9 only operates on the floppy disk drives, hard drives or optical drives. Yet
10 another prior art approach to virus detection performs a checksum on all host
11 programs stored on a system and known to be free from viruses. Thus, if a virus
12 later attaches itself to a host program, the checksum value will be different and
13 the presence of a virus can be detected.

14 Nonetheless, these approaches of the prior art suffer from a number of
15 shortcomings. First, behavior interception is not successful at detecting all
16 viruses because critical operations that may be part of the code for a virus can be
17 placed at locations where such critical operations are likely to occur for the
18 normal operation of programs. Second, most signature scanning is only
19 performed on new inputs from disk drives. With the advent of the Internet and
20 its increased popularity, there are no prior art methods that have been able to
21 successfully scan connections 36 such as those utilized by a gateway node in
22 communicating with other networks. Third, many of the above methods
23 require a significant amount of computing resources, which in turn degrades the

1 overall performance of system. Thus, operating the virus detection programs on
2 every computer becomes impractical. Therefore, the operation of many such
3 virus detection programs is disabled for improved performance of individual
4 machines.

5 Therefore, there is a need for a system and method for effectively detecting
6 and eliminating viruses without significantly effecting the performance of the
7 computer. Moreover, there is a need for a system and method that can detect
8 and eliminate viruses in networks attached to other information systems by way
9 of gateways or the Internet.

10 11 Summary of the Invention

12 The present invention overcomes the limitations and shortcomings of the
13 prior art with an apparatus and method for detecting and eliminating viruses on
14 a computer network. A system including the present invention is a network
15 formed of a plurality of nodes and a gateway node for connection to other
16 networks. The nodes are preferably microcomputers, and the gateway node
17 comprises: a display device, a central processing unit, a memory forming the
18 apparatus of the present invention, an input device, a network link and a
19 communications unit. The memory further comprises an operating system
20 including a kernel, a File Transfer Protocol (FTP) proxy server, and a Simple Mail
21 Transfer Protocol (SMTP) proxy server. The central processing unit, display
22 device, input device, and memory are coupled and operate to execute the
23 application programs stored in the memory. The central processing unit of the

1 gateway node also executes the FTP proxy server for transmitting and receiving
2 files over the communications unit, and executes the SMTP proxy server for
3 transmitting and receiving messages over the communications unit. The FTP
4 proxy server and SMTP proxy server are preferably executed concurrently with
5 the normal operation of the gateway node. The servers advantageously operate
6 in a manner such that viruses transmitted to or from the network in messages
7 and files are detected before the files are transferred into or from the network.
8 The gateway node of the present invention is particularly advantageous because
9 the impact of using the FTP proxy server and SMTP proxy server for the
10 detection of viruses is minimized because only the files leaving or entering the
11 network are evaluated for the presence of viruses and all other "intra" network
12 traffic is unaffected.

13 The present invention also comprises a method for processing a file before
14 transmission into the network and a method for processing a file before
15 transmission from the network. The preferred method for processing a file
16 comprises the steps of: receiving the data transfer command and file name;
17 transferring the file to the proxy server; performing virus detection on the file;
18 determining whether the file contains any viruses; transferring the file from the
19 proxy server to a recipient node if the file does not contain a virus; and
20 performing a preset action with the file if it does contain a virus. The present
21 invention also includes methods for processing messages before transmission to
22 or from the network that operate in a similar manner.

Brief Description of the Drawings

Figure 1 is a block diagram of a prior art information system with a plurality of networks and a plurality of nodes upon which the present invention operates;

Figure 2 is a block diagram of a preferred embodiment for a gateway node including the apparatus of the present invention;

Figure 3 is a block diagram of a preferred embodiment for a memory of the gateway node including the apparatus of the present invention;

Figure 4 is a block diagram of a preferred embodiment for a protocol layer hierarchy constructed according to the present invention compared to the OSI layer model of the prior art;

Figure 5A is a functional block diagram showing a preferred system for sending data files according to a preferred embodiment of the present invention;

Figure 5B is a functional block diagram showing a preferred system for receiving data files according to a preferred embodiment of the present invention;

Figures 6A, 6B and 6C are a flowchart of the preferred method for performing file transfer according to the present invention;

Figure 7 is a functional block diagram showing a preferred system for transmitting mail messages according to a preferred embodiment of the present invention; and

Figures 8A and 8B are a flow chart of a preferred method for sending messages to/from a network.

1

2 Detailed Description Of The Preferred Embodiment

3 The virus detection system and method of the present invention
4 preferably operates on an information system 20 as has been described above
5 with reference to Figure 1. The present invention, like the prior art, preferably
6 includes a plurality of node systems 30 and at least one gateway node 33 for each
7 network 22, 24, 26. However, the present invention is different from the prior
8 art because it provides novel gateway node 33 that also performs virus detection
9 for all files being transmitted into or out of a network. Furthermore, the novel
10 gateway node 33 also performs virus detection on all messages being transmitted
11 into or out of an associated network.

12 Referring now to Figure 2, a block diagram of a preferred embodiment of
13 the novel gateway node 33 constructed in accordance with the present invention
14 is shown. A preferred embodiment of the gateway node 33 comprises a display
15 device 40, a central processing unit (CPU) 42, a memory 44, a data storage device
16 46, an input device 50, a network link 52, and a communications unit 54. The
17 CPU 42 is connected by a bus 56 to the display device 40, the memory 44, the data
18 storage device 46, the input device 50, the network link 52, and the
19 communications unit 54 in a von Neumann architecture. The CPU 42, display
20 device 40, input device 50, and memory 44 may be coupled in a conventional
21 manner such as a personal computer. The CPU 42 is preferably a microprocessor
22 such as an ~~Motorola~~ ^{MOTOROLA} 68040 or ~~Intel Pentium~~ ^{INTEL PENTIUM} or X86 type processor; the display
23 device 40 is preferably a video monitor; and the input device 50 is preferably a

1 keyboard and mouse type controller. The CPU 42 is also coupled to the data
2 storage device 44 such as a hard disk drive in a conventional manner. Those
3 skilled in the art will realize that the gateway node 33 may also be a mini-
4 computer or a mainframe computer.

5 The bus 56 is also coupled to the network link 52 to facilitate
6 communication between the gateway node 33 and the other nodes 30 of the
7 network. In the preferred embodiment of the present invention, the network
8 link 52 is preferably a network adapter card including a transceiver that is
9 coupled to a cable or line 36. For example, the network link 52 may be an
10 ethernet card connected to a coaxial line, a twisted pair line or a fiber optic line.
11 Those skilled in the art will realize that a variety of different networking
12 configurations and operating systems including token ring, ethernet, or arcnet
13 may be used and that the present invention is independent of such use. The
14 network link 52 is responsible for sending, receiving, and storing the signals sent
15 over the network or within the protected domain of a given network. The
16 network link 52 is coupled to the bus 56 to provide these signals to the CPU 34
17 and vice versa.

18 The bus 56 is also coupled to the communications unit 54 to facilitate
19 communication between the gateway node 33 and the other networks.
20 Specifically, the communications unit 54 is coupled to the CPU 42 for sending
21 data and message to other networks. For example, the communications unit 54
22 may be a modem, a bridge or a router coupled to the other networks in a
23 conventional manner. In the preferred embodiment of the present invention,

1 the communications unit 54 is preferably a router. The communications unit 54
2 is in turn coupled to other networks via a media 34 such as a dedicated T-1
3 phone line, fiber optics, or any one of a number of conventional connecting
4 methods.

5 The CPU 42, under the guidance and control of instructions received from
6 the memory 44 and from the user through the input device 50, provides signals
7 for sending and receiving data using the communications unit 54. The transfer
8 of data between networks is broken down into the sending and receiving files
9 and messages which in turn are broken down into packets. The methods of the
10 present invention employ a virus detection scheme that is applied to all transfers
11 of messages and files into or out of a network via its gateway node 33.

12 Referring now to Figure 3, the preferred embodiment of the memory 44
13 for the gateway node 33 is shown in more detail. The memory 44 is preferably a
14 random access memory (RAM), but may also include read-only memory (ROM).
15 The memory 44 preferably comprises a File Transfer Protocol (FTP) proxy server
16 60, a Simple Mail Transfer Protocol (SMTP) proxy server 62, and an operating
17 system 64 including a kernel 66. The routines of the present invention for
18 detecting viruses in file transfers and messages primarily include the FTP proxy
19 server 60 and the SMTP proxy server 62. The FTP proxy server 60 is a routine for
20 controlling file transfers to and from the gateway node 33 via the
21 communications unit 54, and thus controlling file transfers to and from a given
22 network of which the gateway node is a part. The operation of the FTP proxy
23 server 60 is described below in more detail with reference to Figures 5A, 5B, 6A,

1 6B and 6C. Similarly, the SMTP proxy server 62 is a routine for controlling the
2 transfer of messages to and from the gateway node 33, and thus to and from the
3 respective network associated with the gateway node 33. The operation of the
4 SMTP proxy server 62 is described below in more detail with reference to Figure 7
5 8A and 8B. The present invention preferably uses a conventional operating
6 system 28 such as Berkeley Software Distribution UNIX. Those skilled in the art
7 will realize how the present invention may be readily adapted for use with other
8 operating systems such as ~~Macintosh~~ ^{MACINTOSH} System Software version 7.1, DOS ,
9 ~~Windows~~ ^{WINDOWS} or ~~Windows~~ ^{WINDOWS} NT. The memory 44 may also include a variety of
10 different application programs 68 including but not limited to computer drawing
11 programs, word processing programs, and spreadsheet programs. The present
12 invention is particularly advantageous over the prior because it minimizes the
13 impact of virus detection and elimination since the FTP proxy server 60 and
14 SMTP proxy server 62 are preferably only included or installed in the memory 44
15 of the gateway nodes 33. Thus, all data being transferred inside the protected
16 domain of a given network will not be checked because the data packets might
17 not be routed via the gateway node 33.

18 While the apparatus of the present invention, in particular the FTP proxy
19 server 60 and SMTP proxy server 62, has been described above as being located
20 and preferably is located on the gateway node 33, those skilled in the art will
21 realize that the apparatus of the present invention could also be included on a
22 FTP server or a world wide web server for scanning files and messages as they are
23 downloaded from the web. Furthermore, in an alternate embodiment, the

1 apparatus of the present invention may be included in each node of a network
2 for performing virus detection on all messages received or transmitted from that
3 node.

4 As best shown in Figure 4, the CPU 42 also utilizes a protocol layer
5 hierarchy to communicate over the network. The protocol layers of the
6 hierarchy of the present invention are shown in Figure 4 in comparison to the
7 ISO-OSI reference model, for example. The protocol layers 410-426 of the
8 hierarchy of the present invention are similar to the prior art protocol layers for
9 the lower four layers 400-403 including: (1) a physical layer 400 formed of the
10 transmission media 410; (2) a data link layer 401 formed of the network interface
11 cards 411; (3) a network layer 402 formed of address resolution 412, Internet
12 protocol 413 and Internet control message protocol 414; and (4) a transport layer
13 403 formed of the transmission control protocol 415 and a user datagram protocol
14 416. Corresponding to the presentation 405 and session 404 layers, the protocol
15 hierarchy of the present invention provides four methods of communication: a
16 file transfer protocol 417, a simple mail transfer protocol 419, a TELNET protocol
17 419 and a simple network management protocol 420. There are corresponding
18 components on the application layer 406 to handle file transfer 423, electronic
19 mail 424, terminal emulation 425, and network management 426. The present
20 invention advantageously detects, controls and eliminates viruses by providing
21 an additional layer between the application layer 406 and the presentation layer
22 405 for the gateway nodes 33. In particular, according to the hierarchy of the
23 present invention, a FTP proxy server layer 421 and a SMTP proxy server layer

422 are provided. These layers 421, 422 operate in conjunction with the file transfer layer 423 and file transfer protocol 417, and the electronic mail layer 424 and the SMTP protocol layer 418, to process file transfers and messages, respectively. For example, any file transfer requests are generated by the file transfer application 423, first processed by the FTP proxy server layer 421, then processed by the file transfer protocol 417 and other lower layers 415, 413, 411 until the data transfer is actually applied to the transmission media 410. Similarly, any messaging requests are first processed by the SMTP proxy server layer 418, and thereafter processed by the SMTP protocol and other lower layers 415, 413, 411 until the physical layer is reached. The present invention is particularly advantageous because all virus screening is performed below the application level. Therefore, the applications are unaware that such virus detection and elimination is being performed, and these operations are completely transparent to the operation of the application level layers 406. While the FTP proxy server layer 421 and the SMTP proxy server layer 422 have been shown in Figure 4 as being their own layer to demonstrate the coupling effects they provide between the file transfer layer 423 and file transfer protocol 417, and the electronic mail layer 424 and the SMTP protocol layer 418, those skilled in the art will realize that the FTP proxy server layer 421 and the SMTP proxy server layer 422 can also be correctly viewed as being part of the file transfer protocol layer 417 and the SMTP protocol layer 418, respectively, because they are invisible or transparent to the application layer 406.

1 A preferred method of operation and an embodiment for the FTP proxy
2 server 60 will be described focusing on its relationship to and its control of the
3 gateway node 33, and thus, control over access to the medium, line 34, for
4 connections to other networks. The method can best be understood with
5 reference to Figures 5A and 5B, that graphically show the functions performed by
6 an Internet daemon 70, the FTP proxy server 60, and an FTP daemon 78, each of
7 which resides on the gateway node 33. In Figures 5A and 5B, like reference
8 numbers have been used for like parts and the figures are different only in the
9 direction in which the file is being transferred (either from client task 72 to
10 server task 82 or from server task 82 to client task 72). For the sake of clarity and
11 ease of understanding only the data ports are shown in Figures 5A and 5B, and
12 the bi-directional lines represent command or control pathways and are assumed
13 to include a command port although it is not explicitly shown. The operation
14 FTP proxy server 60 will now be described with reference to a file transfer
15 between a client task 72 (requesting machine) and a server task 82 (supplying
16 machine). While it is assumed that the client task 72 (requesting machine) is
17 inside a protected domain and the server task 82 (supplying machine) is outside
18 the protected domain, the invention described below is also used by the gateway
19 node 33 when client task 72 (requesting machine) is outside the protected
20 domain and the server task 82 (supplying machine) is inside the protected
21 domain.

22 Figures 6A-6C are a flowchart of a preferred method for performing file
23 transfers from a controlled domain of a network across a medium 34 to another

1 network (e.g., a file transfer from a node 32 of the second network 24 across the
2 media 34 to a second node 32 of the third network 26). The method begins with
3 step 600 with the client node sending a connection request over the network to
4 the gateway node 33. In step 602, The gateway node 33 preferably has an
5 operating system 64 as described above, and part of the operating system 64
6 includes a fire wall, or program including routines for authenticating users. The
7 gateway node 33 first tries to authenticate the user and decide whether to allow
8 the connections requested, once the request is received. This is done in a
9 conventional manner typically available as part of UNIX. The Internet daemon
10 70 creates an instance of the FTP proxy server 60 and passes the connection to the
11 FTP proxy server 60 for servicing in step 602. The Internet daemon 70 is program
12 that is part of the operating system 64, and it runs in the background. When
13 being run, one of the functions of the Internet daemon 70 is to bind socket ports
14 for many well-known services, such as TELNET, login, and FTP. When a
15 connect request is detected, the Internet daemon 70 constructed in accordance
16 with the present invention, spawns the FTP proxy server 60, which is the server
17 that will actually handle the data transfer. Thereafter, the FTP proxy server 60
18 controls the network traffic passing between the client task 72 and the server task
19 82. Then in step 604, the client node sends a data transfer request and file name,
20 and established a first data port 76 through which the data will be transferred
21 between the FTP proxy server 60 and the client task 72. In step 606 the data
22 transfer request and file name are received by the FTP proxy server 60. In step
23 608, the FTP proxy server 60 determines whether the data is being transferred in

1 an outbound direction (e.g., the file is being transferred from the client task 72 to
2 the server task 82). This can be determined by the FTP proxy server 60 by
3 comparing the data transfer request. For example, if the data transfer request is
4 the STOR command then the data is being transferred in an outbound direction;
5 and if the data transfer request is the RETR command then the data is not being
6 transferred in an outbound direction.

7 If the data is being transferred in an outbound direction, then the method
8 transitions from step 608 to step 610. Referring now to Figure 6B in conjunction
9 with Figure 5A, the process for transferring data out of the protected domain of
10 the network is described in more detail. In step 610, the FTP proxy server 60
11 determines whether the file to be transferred is of a type that can contain viruses.
12 This step is preferably performed by checking the extension of the file name. For
13 example, .txt, .bmd, .pcx and .gif extension files indicate that the file is not likely
14 to contain viruses while .exe, .zip, and .com extension files are of the type that
15 often contain viruses. If the file to be transferred is not of a type that can contain
16 viruses, then the method continues in step 612. In step 612, a second data port 80
17 is established and the data transfer request & the file are sent from the FTP proxy
18 server 60 to the FTP daemon 78 so that the file can be sent to the server task 82.
19 The FTP daemon 78 is a program executed by the gateway node 33 that
20 communicates the transfer commands to the server task 82, establishes a third
21 port 84 for sending the file including binding the server task 82 and FTP daemon
22 78 to the third port 84, and transmits the file to the server task 82. Once
23 transmitted, the method is complete and ends. However, if it is determined in

1 step 610 that the file to be transferred is of a type that can contain viruses, the
2 method proceeds to step 614. In step 614, the FTP proxy server 60 transfers the
3 file from the client to the FTP proxy server 60 through the first port 76, and in
4 step 616, the file is temporarily stored at the gateway node 33. Then in step 618,
5 the temporarily stored file is analyzed to determine if it contains viruses. This is
6 preferably done by invoking a virus-checking program on the temporarily stored
7 file. For example, a program that performs a version of signature scanning virus
8 detection such as PC-Cillin manufactured and sold by Trend Micro Devices
9 Incorporated of Cupertino, California may be used. However, those skilled in
10 the art will realize that various other virus detection methods may also be used
11 in step 618. In step 620, output of the virus checking program is preferably
12 echoed to the user/client task 72 by the FTP proxy server 60 as part of a reply
13 message. Next in step 622, the method determines whether any viruses were
14 detected. If no viruses are detected, the method continues in step 612 and
15 transmits the file as has been described above. However, if a virus is detected,
16 the present invention advantageously allows the FTP proxy server 60 to respond
17 in any number of a variety of ways. The response of the FTP proxy server 60 is
18 determined according to user's needs and wants as specified in a configuration
19 file. This configuration file is preferably fully modifiable according to input from
20 the user and stored in memory 44. For example, some options the user might
21 specify are: 1) to do nothing and transfer the file; 2) to delete or erase the
22 temporary file and do not transfer the file; or 3) to rename the file and store it in
23 a specified directory on the gateway node 33 and notify the user of the new file

1 name and directory path which can used to manually request the file from the
2 system administrator. Those skilled in the art will realize that there are variety
3 of other alternatives that users might specify, and steps 624, 626, and 628 are
4 provided only by way of example. Next in step 624, the configuration file is
5 retrieved to determine the handling of the temporary file. In step 626, the FTP
6 proxy server 60 determines if it is to ignore the existence of a virus and a
7 continue the transfer. If so, the method continues in step 612 where the file is
8 passed to the FTP daemon 78 and the temporary file is deleted. If not the method
9 continues to step 628 where either the file is deleted and not sent to the server
10 task 82, and the temporary file is erased from the gateway node 33; or the file is
11 renamed and stored in a specified directory on the gateway node 33 and the user
12 is notified of the new file name and directory path which can used to manually
13 request the file from the system administrator, and the temporary file is erased
14 the gateway node 33. The action taken in step 628 depends on the configuration
15 settings as determined in step 624. After step 628, the method ends. As can be
16 seen from Figure 5A, the path for the file is from client task 72 through the first
17 data port 76 to the FTP proxy server 60, then to the FTP daemon 78 through the
18 second data port 80 and finally to the server task 82 through the third data port
19 84.

20 Referring back to step 608 of Figure 6A, if the data is not being transferred
21 in an outbound direction, then the method transitions from step 608 to step 640.
22 Referring now to Figure 6C in conjunction with Figure 5B, the process for
23 transferring data into the protected domain of the network is described in more

1 detail. In step 640, the FTP proxy server 60 next sends the data transfer request
2 and file name first to the FTP daemon 78 and then on to the server task 82. In
3 step 642, a second port 80 is established between the FTP proxy server 60 and the
4 FTP daemon 78. Then a third data port 84 is established between the FTP
5 daemon 78 and the server task 82. Both ports 80, 84 are established similar to the
6 establishment of the first port 76. The FTP daemon 78 will request and obtain the
7 third port 84 from the Internet daemon 70, and send a port command to the
8 server task 82 including an address for the third port 84. The server task 82 will
9 then connect to the third port 84 and begin the data transfer in step 644. The FTP
10 daemon 78 in turn sends the file to the FTP proxy server 60. Next in step 646, the
11 FTP proxy server 60 determines whether the file to be transferred is of a type that
12 can contain viruses. This is done the same was as described above with reference
13 to step 610. If the file to be transferred is not of a type that can contain viruses,
14 then the method continues in step 648 where the file is transferred from the FTP
15 proxy server 60 through the first port 76 to the client task 72, then the method is
16 complete and ends. On the other hand, if the file to be transferred is a type that
17 can contain viruses, the method in step 650 temporarily stores the file at the
18 gateway node. Then in step 652, the temporarily stored file is analyzed to
19 determine if it contains viruses. The analysis here is the same as step 618. In
20 step 652, the output of the virus checking program is preferably echoed to the
21 client task 72 by the FTP proxy server 60 as part of a reply message. Next in step
22 656, the method determines whether any viruses were detected. If no viruses are
23 detected, the method continues in step 648 as has been described above.

1 However, if a virus is detected, the present invention retrieves the configuration
2 file to determine the handling of the temporary file. In step 660, the FTP proxy
3 server 60 determines if it is to ignore the existence of a virus and a continue the
4 file transfer. If so the method continues in step 648 where the file is passed to the
5 client task 72 and the temporary file is erased. If not the method continues to
6 step 662 where the temporary file is erased, and the file is either deleted and not
7 sent to the client task 72 or the file is renamed, stored on the gateway node 33,
8 and the client task 72 is notified of new name and path so that the file may be
9 manually retrieved by the system administrator. The method then ends. As can
10 be seen from Figure 5B, the data transfer request is passed from the client task 72,
11 to the FTP proxy server 60, then to the FTP daemon 78, and to the server task 82
12 which in response sends the file through the third port to the FTP daemon 78,
13 and through the second port 80 on to the FTP proxy server 60, and finally
14 through the first port 76 to the client task 72.

15 Referring now to Figures 7, 8A and 8B, the operation of the SMTP proxy
16 server 62 will now be described. The SMTP proxy server 62 controls the only
17 other entry channel through which data, and therefore viruses, can enter the
18 protected domain of a given network. The SMTP proxy server 62 is preferably a
19 program that resides on the gateway node 33, and controls and handles all
20 transfers of electronic messages or mail in and out of the network through the
21 communications unit 54 and media 34. While the SMTP proxy server 62 will
22 now be described with reference to the transfer of a mail message from a client
23 task 92 within the protected domain of the network to a server task 102 at a node

on a different network outside the protected domain, those skilled in the art will understand how the SMTP proxy server 62 handles incoming mail messages in the same way. All mail messages are handled by the SMTP proxy server 62 in the same way and only the designation of which node 32 is the server and which is the client change depending on the direction the message is being sent from the perspective of the gateway node 33. Since mail messages are passed using the command pathways between nodes, only these pathways are shown in Figure 7. For ease of understanding, the command ports have not been shown in Figure 7, but will be discussed below in the relevant steps of the preferred method.

Referring now to Figure 8A, the preferred method of the present invention for sending electronic mail begins in step 802 with the spawning or running the SMTP proxy server 62. Next in step 804, a first command port 96 for communication between the client task(s) 92 and the SMTP proxy server 62 is created. The address of the first port 96 along with a port command is provided to the SMTP proxy server 62. Then in step 806, the SMTP proxy server 62 is bound to the first port 96 to establish a channel for sending a mail message between any client tasks and the SMTP proxy server 62. Next in step 808, the SMTP proxy server 62 spawns a SMTP daemon 98 or SMTP server. The SMTP daemon 98 is preferably the existing program "sendmail" that is part of the BSD UNIX operating system. This is particularly advantageous because it reduces the amount of code that needs to be written and assures compatibility with the lower layers of the OSI reference model. Then in step 810 a second command port is created for communication between the SMTP proxy server 62 and the SMTP

1 daemon 98. In step 812, the SMTP daemon 98 is bound to the second command
2 port for communication with the SMTP proxy server 62. Actually, the present
3 invention binds the SMTP daemon 98 to the appropriate port, namely the second
4 port by redefining the bind function in a shared library that is part of the
5 operating system 64. The present invention advantageously exploits the fact that
6 the SMTP daemon 98 (sendmail programs on most UNIX systems) are
7 dynamically linked. The present invention utilizes a shared library which
8 redefines the system call bind() and forces sendmail to link with the redefined
9 version of the bind() call when executed. If the redefined version of the bind()
10 call determines the SMTP daemon 98 (sendmail program) is trying to bind to the
11 first command port (the smtp port), it will return to it a socket whose other end
12 is the SMTP proxy server 62 (a socket to the second command port). Next in step
13 800, the client task 92 request a connection from the SMTP proxy server and is
14 directed to used the first command port for communication. Then in step 818,
15 the message is transmitted from the client task 92 through the first command
16 port to the SMTP proxy server 62.

17 Referring now to Figure 8B, the method continues in step 820 with the
18 SMTP proxy server 62 scanning the message body and checking for any portions
19 that are encoded. The present invention preferably scans the message for
20 portions that have been encoded with an "uuencoded" encoding scheme that
21 encodes binary data to ASCII data. "Uuencoded" portions of messages usually
22 start with a line like "begin 644 filename," and end with a line like "end." The
23 existence of such encoded portions suggests the possibility that a file may contain

1 viruses. This scanning for "uuencoded" portions is just one of many scanning
2 techniques that may be used, and those skilled in the art will realize that the
3 present invention could be modified to scan for other encoded portions such as
4 those encoded according to other schemes such as mime. Next in step 822, the
5 SMTP proxy server 62 determines whether the message includes any encoded
6 portions. If the message does not include any encoded portions, the SMTP proxy
7 server 62 transmits the message through the second command port to the SMTP
8 daemon 98 in step 824. Next in step 814, the SMTP daemon 98 creates a third
9 command port for communication between the SMTP daemon 98 and the server
10 task 102. Then in step 816 the server task 102 is bound to the third command
11 port to establish communication between the server task 102 and the SMTP
12 daemon 98. Those skilled in the art will realize that if the server task 102 resides
13 on the gateway node 33, then steps 814 and 816 are not needed and may be
14 omitted since no further transfer of data across the network is needed. Then the
15 SMTP daemon 98 transmits the message through the third command port to the
16 server task 102 in step 826 thereby completing the method.

17 On the other hand if in step 822 it is determined the message does include
18 encoded portions, the SMTP proxy server 62 stores each of the encoded portions
19 of the message in its own temporary file at the gateway node 33 in step 828. For
20 example, if a message included three encoded portions, each encoded portion
21 will be stored in a separate file. Then in step 830, each of the encoded portions
22 stored in its own file is individually decoded using uudecode program, as will be
23 understood by those skilled in the art. Such decoding programs known in the art

1 convert the ASCII files back to their original binary code. Next in step 832, the
2 SMTP proxy server 62 calls and executes a virus-checking program on each
3 message portion stored in its temporary file(s). Then in step 834, the SMTP proxy
4 server 62 determines whether any viruses were detected. If no viruses are
5 detected, the method continues to steps 824, 814, 816 and 826 as has been
6 described above. However, if a virus is detected, the present invention
7 advantageously allows the SMTP proxy server 62 to respond in any number of a
8 variety of ways, just as the FTP proxy server 60. The response of the SMTP proxy
9 server 62 is also determined by the according to user's needs and wants as
10 specified in a configuration file. This configuration file is preferably fully
11 modifiable according to input from the user. The configuration for virus
12 handling is determined in step 836. This could be done by retrieving and reading
13 the configuration file or simply retrieving the configuration data already stored
14 in memory 44. Then in step 838, the action to be taken is determined from the
15 configuration settings. For example, some options the user might specify are: 1)
16 to do nothing and transfer the mail message unchanged; 2) to transfer the mail
17 message with the encoded portions that have been determined to have viruses
18 deleted from the mail message; 3) rename the encode portions of the message
19 containing viruses, store the renamed portions as files in a specified directory on
20 the SMTP proxy server 62 and notify the user of the renamed files and directory
21 path which can used to manually request the file from the system administrator;
22 or 4) writing the output of step 832 into the mail message in place of the
23 respective encoded portions and sending that mail message in steps 824 and 826.

1 Once the action to be performed has been determined from examination of the
2 configuration file, the specified action is taken in step 840, the transformed
3 message is transmitted, the temporary file is erased, and the method ends. For
4 example, if a message has three encoded portions, two encoded portions contain
5 viruses, and the configuration file indicates that virus containing portions are to
6 be deleted, then the method of the present invention would send a transformed
7 message that was the same as the original message, but with the two encoded
8 portions containing viruses deleted, to the server task 102.

9 While the present invention has been described with reference to certain
10 preferred embodiments, those skilled in the art will recognize that various
11 modifications may be provided. For example, the preferred operation of the
12 present invention specifies that the FTP proxy server 60 determine whether the
13 file type is one that can contain a virus (Steps 610 and 646). However, alternate
14 embodiments can omit these steps and simply temporarily store and scan all files
15 being transferred for viruses. Likewise the SMTP proxy server 60 may, in
16 alternate embodiments, omit the step 822 of determining whether the message is
17 encoded and temporarily store and scan all message being transmitted for
18 viruses. Furthermore, while the invention has been described above as
19 temporarily storing the file or message at the gateway node in a temporary file,
20 this step could be omitted in the determination of whether a file includes a virus
21 were done as the file was being transferred from the client node to the gateway
22 node. These and other variations upon and modifications to the preferred

1 embodiment are provided for by the present invention which is limited only by
2 the following claims.